REMARKS/ARGUMENTS

Favorable reconsideration of this application as presently amended and in light of the following discussion is respectfully requested.

Claims 1-2, 4-7, 9-12 and 14-15 are presently active, Claims 3, 8 and 13 have been previously canceled without prejudice, and Claims 1, 6 and 11 are amended to clarify the claimed subject matter. No new matter is added.

In the outstanding Office Action, the specification was objected to because of informalities; and Claims 1-2, 4-7, 9-12 and 14-15 were rejected under 35 U.S.C. § 102(e) as being anticipated by Silva et al. (U.S. Pat. No. 6,976,210).

Regarding the objection to the specification, the specification is amended as suggested in the outstanding Office Action. Therefore, it is respectfully submitted that the objection to the specification is overcome.

Regarding the rejection of Claims 1-2, 4-7, 9-12 and 14-15, Applicants respectfully submit that the rejection is overcome because, in Applicants' view, amended independent Claims 1, 6 and 11 patentably distinguish over Silva et al. as discussed below.

Amended Claim 1 recites, *inter alia*, "converting the document structure generated by the inserting step into a desired document structure ***by carrying out at least one of elimination, addition and rearrangement of one or the plurality of partial and second elements***, according to ranges of the second document to be converted including the partial documents inserted by the inserting step and ***identification information of a file describing a conversion rule*** for converting the document structure into the desired document structure, the ranges of the second document to be converted and ***the identification information*** described by the specific markup language and ***included in the second document***."

According to a non-limiting example of the invention recited in Claim 1 (see the original specification at page 36, lines 2-10), an XML-P'z document shown in Fig. 16 of the present application is for converting an original document structure of one textbook data, which is contained therein (see, for example, the "textbook" element E1 in Fig. 16), and an original document structure of another textbook data, which is contained in a web document stored at http://www.xxx.com/booklist.xml (see, for example, the "pz:targets" element E2 in Fig. 16), into a common book format (*i.e.*, a desired document structure) based on a conversion rule described in an XSLT document "textbook-book.xsl." Thus, a composed web document (XML document) W1 is outputted. This operation comprises the following steps:

(1) an interpreter 102 carries out an interpretation processing of the "pz:targets" element E2 in an XML-DOM tree shown in Fig. 17; and

(2) the interpreter 102 carries out an interpretation processing of the "pz:convert" element E3 in an XML-DOM tree shown in Fig. 18.

In the step (1), a target command processor 122 changes the XML-DOM tree shown in Fig. 17 into the XML-DOM tree shown in Fig. 18 by replacing the "pz:targets" element E2 with a new element E2', and generates a document structure containing the original document structures expressed by the "textbook" element E1 and the new element E2'. A plurality of textbook data existing in the web document stored at http://www.xxx.com/booklist.xml are inserted as the XML-DOM trees shown in Fig. 17.

In the step (2), the interpretation processing of the "pz:convert" element E3 is carried out by using the XSLT document shown in Fig. 19. The XSLT document describes a conversion rule for converting an "author" element, a "publication" element and a "price" element of each textbook data (*i.e.*, the original document structure) into a "title" element, a "price" element and an "author" element (*i.e.*, the desired document structure), respectively (see Figs. 17, 18 and 20). Thus, in this example, the "publication" element is eliminated, the "title" element is added, and the "price" element and the "author" element are rearranged.

That is, the XSLT document describes the conversion rule for carrying out at least one of elimination, addition and rearrangement of one or the plurality of partial and second elements in the generated document structure, as recited in amended Claim 1.

On the other hand, Silva et al. describes generating an equivalent well-formed version of a source Web page from which a clipping is to be extracted (Silva et al. at col. 7, lines 17-52). ***The term "well-formed" means that elements delimited by a start-tag and an end-tag nest properly within each other in a document containing the elements*** (see the attached document: "2.1 Well-Formed XML Documents" in Extensible Markup Language (XML) 1.0 (Fourth Edition) http://www.w3.org/TR/REC-xml/#sec-well-formed).

For example, the extracted document has the following original document structure:

```
<div>
<h1> SONNY
<h2> Digital Camera S12000
<span> Price $300, Resolution 500M, Color Black </span>
</div>
```

Since the "h1" tag and the "h2" tag are not closed in the original document structure, the system of Silva et al. closes these tags so that the document structure has a well-formed structure as follows:

```
<div>
<h1> SONNY </h1>
<h2> Digital Camera S12000 </h2>
<span> Price $300, Resolution 500M, Color Black </span>
</div>
```

As such, this conversion standardizes ambiguous grammars of the original document structure. However, elimination, addition and rearrangement of the elements, such as "SONNY," "Digital Camera S12000," "Price," "Resolution" and "Color" are not carried out. That is, Silva et al. does not teach elimination, addition and rearrangement of those elements by the conversion.

Thus, Silva et al. fails to teach or suggest "converting the document structure generated by the inserting step into a desired document structure ***by carrying out at least one of elimination, addition and rearrangement of one or the plurality of partial and second elements***, ...," as recited in Claim 1.

In addition, the outstanding Office Action states that Silva et al. teaches that by parsing the displayed page (*first document*), the parser generates a document object model of the source Web page in which element are uniquely identifiable (*identification information*) (Office Action at page 6, the last line through page 7, line 2).

However, the identification information recited in Claim 1 is that ***of a file describing a conversion rule***, not that of the elements of the document object model of the source Web page. Although Claim 1 recites that the identification information of a file describing a conversion rule is included in the second document, which the outstanding Office Action asserts corresponds to the Web view of Silva et al., Silva et al. does not teach that identification information of a file describing a conversion rule is included in the Web view.

Thus, Silva et al. fails to teach or suggest "..., according to ranges of the second document to be converted including the partial documents inserted by the inserting step and ***identification information of a file describing a conversion rule*** for converting the document structure into the desired document structure, the ranges of the second document to be converted and ***the identification information*** described by the specific markup language and ***included in the second document***," as recited in Claim 1.

Likewise, Silva et al. fails to teach or suggest "a conversion unit configured to convert the document structure generated by the insertion unit into a desired document structure by carrying out at least one of elimination, addition and rearrangement of one or the plurality of partial and second elements, according to ranges of the second document to be converted including the partial documents inserted by the insertion unit and identification information of a file describing a conversion rule for converting the document structure into the desired document structure, the ranges of the second document to be converted and the identification information described by the specific markup language and included in the second document," as recited in Claim 6.

Likewise, Silva et al. fails to teach or suggest "third computer program codes for causing the computer to convert the document structure generated by the second computer program codes into a desired document structure by carrying out at least one of elimination, addition and rearrangement of one or the plurality of partial and second elements, according to ranges of the second document to be converted including the partial documents inserted by the second computer program codes and identification information of a file describing a conversion rule for converting the document structure into the desired document structure, the ranges of the second document to be converted and the identification information described by the specific markup language and included in the second document," as recited in Claim 11.

Accordingly, independent Claims 1, 6 and 11 patentably distinguish over Silva et al. Therefore, Claims 1, 6 and 11 and the pending Claims 2, 4-5, 7, 9-10, 12 and 14-15 dependent directly or indirectly from Claims 1, 6 and 11 are believed to be allowable.

Consequently, in view of the present amendment and in light of the above discussions, it is believed that the outstanding rejection is overcome, and the application as amended herewith is believed to be in condition for formal allowance. An early and favorable action to that effect is respectfully requested.

Respectfully submitted,

OBLON, SPIVAK, McCLELLAND,
MAIER & NEUSTADT, P.C.

[illegible]

Eckhard H. Kuesters
Attorney of Record
Registration No. 28,870

Customer Number
22850

Tel: (703) 413-3000
Fax: (703) 413 -2220
(OSMMN 06/04)

EHK/TY:pta
I:\ATTY\TY\AMEND-RESPONSES\217398\217398 AM DUE APRIL 22 2007.DOC

representation in both strings. No case folding is performed. (Of strings and rules in the grammar:) A string matches a grammatical production if it belongs to the language generated by that production. (Of content and content models:) An element matches its declaration when it conforms in the fashion described in the constraint **[VC: Element Valid]**.]

for compatibility

[Definition: Marks a sentence describing a feature of XML included solely to ensure that XML remains compatible with SGML.]

for interoperability

[Definition: Marks a sentence describing a non-binding recommendation included to increase the chances that XML documents can be processed by the existing installed base of SGML processors which predate the WebSGML Adaptations Annex to ISO 8879.]

2 Documents

[Definition: A data object is an **XML document** if it is well-formed, as defined in this specification. In addition, the XML document is valid if it meets certain further constraints.]

Each XML document has both a logical and a physical structure. Physically, the document is composed of units called entities. An entity may refer to other entities to cause their inclusion in the document. A document begins in a "root" or document entity. Logically, the document is composed of declarations, elements, comments, character references, and processing instructions, all of which are indicated in the document by explicit markup. The logical and physical structures MUST nest properly, as described in **4.3.2 Well-Formed Parsed Entities**.

2.1 Well-Formed XML Documents

[Definition: A textual object is a **well-formed** XML document if:]

1. Taken as a whole, it matches the production labeled document.
2. It meets all the well-formedness constraints given in this specification.
3. Each of the parsed entities which is referenced directly or indirectly within the document is well-formed.

Document

```
[1]  document  ::=  prolog element Misc*
```

Matching the document production implies that:

1. It contains one or more elements.
2. [Definition: There is exactly one element, called the **root**, or document element, no part of which appears in the content of any other element.] For all other elements, if the start-tag is in the content of another element, the end-tag is in the content of the same element. More simply stated, the elements, delimited by start- and end-tags, nest properly within each other.

[Definition: As a consequence of this, for each non-root element `C` in the document, there is one other element `P` in the document such that `C` is in the content of `P`, but is not in the content of any other element that is in the content of `P`. `P` is referred to as the **parent** of `C`, and `C` as a **child** of `P`.]

2.2 Characters

[Definition: A parsed entity contains **text**, a sequence of characters, which may represent markup or character data.] [Definition: A **character** is an atomic unit of text as specified by ISO/IEC 10646:2000 [ISO/IEC 10646]. Legal characters are tab, carriage return, line feed, and the legal characters of Unicode and ISO/IEC 10646. The versions of these standards cited in **A.1 Normative References** were current at the time this document was prepared. New characters may be added to these standards by amendments or new editions. Consequently, XML processors MUST accept any character in the range specified for Char.]

Character Range

[2]	`Char`	::=	`#x9 | #xA | #xD | [#x20-#xD7FF] | [#xE000-#xFFFD] | [#x10000-#x10FFFF]`	*/* any Unicode character, excluding the surrogate blocks, FFFE, and FFFF. */*

The mechanism for encoding character code points into bit patterns may vary from entity to entity. All XML processors MUST accept the UTF-8 and UTF-16 encodings of Unicode 3.1 [Unicode3]; the mechanisms for signaling which of the two is in use, or for bringing other encodings into play, are discussed later, in **4.3.3 Character Encoding in Entities**.

Note:

Document authors are encouraged to avoid "compatibility characters", as defined in section 6.8 of [Unicode] (see also D21 in section 3.6 of [Unicode3]). The characters defined in the following ranges are also discouraged. They are either control characters or permanently undefined Unicode characters:

```
[#x7F-#x84], [#x86-#x9F], [#xFDD0-#xFDDF],
[#x1FFFE-#x1FFFF], [#x2FFFE-#x2FFFF], [#x3FFFE-#x3FFFF],
[#x4FFFE-#x4FFFF], [#x5FFFE-#x5FFFF], [#x6FFFE-#x6FFFF],
[#x7FFFE-#x7FFFF], [#x8FFFE-#x8FFFF], [#x9FFFE-#x9FFFF],
[#xAFFFE-#xAFFFF], [#xBFFFE-#xBFFFF], [#xCFFFE-#xCFFFF],
[#xDFFFE-#xDFFFF], [#xEFFFE-#xEFFFF], [#xFFFFE-#xFFFFF],
```